Enhancing IEEE 802.11 MAC in congested environments

Imad Aad, Qiang Ni, Chadi Barakat, Thierry Turletti

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Enhancing IEEE 802.11 MAC in congested environments

Imad Aad, Qiang Ni, Chadi Barakat, Thierry Turletti

N° 4879

July 2003

THÈME 1

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--4879--FR+ENG

Enhancing IEEE 802.11 MAC in congested environments

Imad Aad, Qiang Ni, Chadi Barakat, Thierry Turletti

Thème 1 — Réseaux et systèmes
Projet Planète

Rapport de recherche n° 4879 — July 2003 — 30 pages

Abstract: IEEE 802.11 is the most deployed wireless local area networking standard nowadays. It uses carrier sense multiple access with collision avoidance (CSMA/CA) to resolve contention between nodes. Contention windows (CW) change dynamically to adapt to the contention level: Upon each collision, a node doubles its CW to reduce further collision risks. Upon a successful transmission, the CW is reset, assuming that the contention level has dropped. However, contention level is more likely to change slowly, and resetting the CW causes new collisions and retransmissions before reaching the optimal value again. This wastes bandwidth and increases delays. In this report we analyze simple slow CW decrease functions and compare their performances to the legacy standard. We use simulations and mathematical modeling to show their considerable outperformance at all contention levels and transient phases.

Key-words: Wireless communications, MAC, CSMA/CA, IEEE 802.11, simulations, Markov chains.

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles, BP 93, 06902 Sophia Antipolis Cedex (France)
Téléphone : +33 4 92 38 77 77 — Télécopie : +33 4 92 38 77 65

Amélioration de IEEE 802.11 dans des environnements congestionnés

Résumé : IEEE 802.11 est le standard de réseaux locaux sans-fil le plus déployé de nos jours. Il utilise CSMA/CA pour résoudre la contention entre transmetteurs. La fenêtre de contention (CW) change dynamiquement pour s'adapter à l'état du réseau; après chaque collision une station double son CW pour réduire les risques de futures collisions. La station remet son CW à sa valeur initiale après chaque bonne transmission, en supposant que le niveau de congestion a diminué. Cependant, très souvent, le niveau de congestion change lentement, et la remise du CW à sa valeur initiale cause de nouvelles collisions et de nouvelles retransmissions avant d'atteindre la bonne valeur de nouveau. Ceci dégrade considérablement les performances (débits, délais etc.). Dans ce rapport nous proposons des fonctions de décrémentation "lentes" des CWs, et nous comparons leurs performances à celles du standard actuel. Nous les analysons mathématiquement et par simulation pour montrer les gains considérables qu'elles peuvent apporter, et ceci quelque soit le niveau de congestion ainsi que durant les phases transitoires.

Mots-clés : Communications sans-fil, couche MAC, CSMA/CA, IEEE 802.11, simulations, chaines de Markov.

1 Introduction

Wireless access networks are experiencing a huge success similar to that of the deployment of the Internet a decade ago. Wireless devices are used almost everywhere to provide cheap, mobile and easy to deploy networks, with or without access to wired infrastructures like the Internet. Wireless access networks can be grouped into two categories: Centralized or distributed. Centralized architectures are mainly controlled by a coordinator (e.g. Access Point (AP)) that grants access to the wireless nodes in its area in a contention free manner. On the other hand, distributed architectures have no central coordinators: All nodes contend to access the channel using a distributed function. The efficiency of this function can be measured by the throughput it achieves and the delays frames observe before they get successfully received.

IEEE 802.11 [1, 2, 3] is the most deployed wireless local area network (LAN) standard nowadays. It supports two access functions, one is centralized at the AP, the other is distributed. The distributed coordination function (DCF) is based on carrier sense multiple access (CSMA)[4] with collision avoidance (CA). Using CSMA/CA, each node differs its transmission to a random time in the future and senses the channel before trying to transmit. Upon each collision, notified by the absence of acknowledgment (ACK) from the destination, the node increases the bound of the random deferring time, called contention window (CW). Increasing the CW reduces the risk of further collisions, assuming the number of contending nodes is high. Nodes may optionally use *request to send / clear to send* (RTS/CTS) frames to reserve the channel before the actual data-ACK frame transmissions. Upon each successful transmission, a node resets its CW to CW_{min} and contends again with low CW values.

Our work in this paper aims to enhance this last point: Upon a successful transmission, a wireless node resets its CW, therefore it takes the risk of experiencing the same collisions and retransmissions until it reaches high CW values again, wasting time and bandwidth. Assuming the number of contending terminals changes slowly, this risk is likely to be high. We propose slow CW decrease (SD) functions and evaluate their performance by comparing them to the IEEE 802.11 standard, in different simulation scenarios. Simulations and mathematical models show that these functions outperform the legacy standard in terms of throughput, delays, jitter and power consumption.

The next section presents the motivations and related work. Section 3 introduces the approach of slow CW decrease (SD) and evaluates its performances from the throughput, delay and jitter point of view, using simulations and mathematical models. Section 4 introduces and analyzes another performance metric, the *settling time* of SD. Section 5 explores the fairness properties of the proposed SD scheme, then section 6 analyzes its energy savings. Section 7 analyzes the channel noise effect on the mechanism and section 8 concludes this paper.

2 Motivations and related work

In a congested environment, a node has no knowledge of the number of contending terminals. However, it adapts its CW to the current congested level by doubling its CW upon each collision, and resetting it upon a successful transmission. Doubling the CW assumes a higher congestion level and therefore the need to increase the CW. When a node increases its CW, it reduces the chances of simultaneous transmissions with other nodes, at the cost of more backoff overhead. This reduces collisions and the corresponding retransmission times, which improves the throughput. When a node succeeds to transmit a frame, it assumes that the congestion level has dropped, and therefore it resets its CW to CW_{min}.

But, when a node succeeds to transmit a frame at a given CW_i, this does not correspond to a congestion level decrease, but to a convenient CW value. Therefore the CW value must be kept large as long as the congestion level remains the same. By resetting the CW, a node takes the risk of experiencing the same collisions and retransmissions "from scratch" until it reaches convenient (high) CW values again, wasting time and bandwidth.

To adapt the CW to possible congestion level drops, we should consider decreasing the CW upon successful transmissions. However, since congestion level is not likely to drop suddenly, we should consider slow CW decrease (SD) functions.

Intuitively, the advantage of SD functions is more collision avoidance during congestion, leading to less collisions and retransmissions, which increases throughput and decreases delays. The drawback is keeping high CW values after congestion level drops, increasing the overhead and decreasing the throughput. This inconvenience is of small importance compared to the advantage of SD, since it is very unlikely that the congestion level drops quickly to small values. In the following sections we propose SD functions and evaluate their performances by comparing them to the actual standard, in different scenarios.

The smooth CW decrease was first introduced in [5], among several other extensions to CSMA and MACA (like backoff copying and per-flow backoff counters). The main idea was to increase the CW at each collision by multiplying it by 1.5, and to linearly decrease it by 1 at each successful frame transmission. The approach was called MILD (multiplicative increase, linear decrease), and did not explore the effect of other decrease (or increase) factors on efficiency. Furthermore, short simulation results were shown for communications between the wireless terminals (WT) and the AP only.

In [6], the smooth CW decrease was considered, but from the fairness enhancement point of view. [6] tries to establish local utility functions in order to achieve system-wide fairness, with no explicit global coordination. Then, it “translates” a given fairness model into a corresponding backoff-based collision resolution algorithms that probabilistically achieve the fairness objective. These algorithms include different backoff increase/decrease factors.

[6] tries to enhance the fairness properties of IEEE 802.11, MACAW [5] and CB-Fair proposed in [7]. Always aiming to establish fair contention algorithms, [7] uses smooth CW increase and decrease functions. Each station i contends to access the channel in order to send a frame to station j with a probability p_{ij}, computed in two ways using time-based and connection-based methods. These methods are pre-established using information broadcast by each station such as the number of logical connections and the contention time.

One can also find some similarites between working for fairness on the MAC sublayer and working on fairness on the transport layer. TCP Reno [8] uses additive increase and multiplicative decrease based on [9] in order to attain fairness among flows.

In this paper we aim to investigate the CW decrease functions from the data rate, delay, response time, fairness and power saving efficiency point of view. We consider general network topologies and schemes to validate our analysis.

3 Throughput, delay and jitter analysis

Consider 100 wireless terminals (WTs) uniformly distributed in a 100x100m square area communicating with each other two by two (50 flows). All nodes are within the range of each other, hence no routing protocol is used.

We start the simulation at time $t = 40$ seconds (s). We increase the number of active flows by one every two seconds. Each transmitting WT sends 1050-byte CBR packets every 5ms (providing full data rate of 1.6 Mbps). At $t = 150s$, all traffic sources stop except one. At $t = 260s$, all sources stop sending data.

Optimally, when the number of contending flows n increases, each flow would get $1/n$ of the available data rate. However, due to the increasing collisions, frames would be corrupted, not acknowledged and retransmitted, which would decrease the actual throughput observed by each flow.

The dashed curve in Fig. 1 shows how the total throughput, averaged over one-second intervals, decreases as the number of contending flows increases (e.g. during $t = 40s \rightarrow 150s$). In fact, after each collision, the source has to wait for a timeout to realize that the frame collided, increases its CW (to reduce further collision risks) then retransmits the frame. After a successful transmission the source resets its CW.

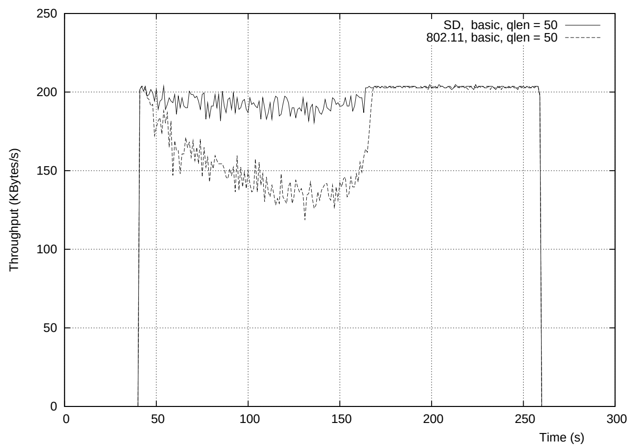

Figure 1: Total throughput comparison, without RTS/CTS.

As a node resets its CW after a successful transmission, it "forgets" about the collision experience it had. If all WTs keep transmitting with the same data rate, most probably the new transmission will observe contention and collisions as before. This can be avoided by keeping some history on the observed collisions: Instead of resetting the CW to CW_{min}, we set the CW to 0.9 times its previous value (lower bounded by CW_{min}, i.e. $CW_{new} = max\{CW_{min}, 0.9 \times CW_{prev}\}$). The solid curve in Fig. 1 shows the considerable throughput enhancement we get (up to 37%), especially with high number of active flows (at $t = 150s$). When we decrease the CW slowly, we waste more backoff time in favor of collision avoidance. Furthermore, throughput is more stable, due to lower/smoother variations of CW values.

SD is a tradeoff between wasting some backoff time and risking a collision followed by the whole frame retransmission. Since the time induced by the latter is much larger than the backoff time, SD is much better on average. The average overhead due to backoff and retransmissions can be written as:

$$E[overhead] = O_{bkof}(j) \times (1 - P_{col}) + O_{retx+bkof} \times P_{col}$$

where P_{col} is the probability of a collision, $O_{bkof}(j)$ is the overhead due to backoff time at stage j of the successful transmission, and

$$O_{retx+bkof} = \sum_{i=1}^{r}(O_{bkof}(i) + T_{data})$$

is the overhead due to retransmissions and their corresponding backoffs, r being the number of retransmissions until a successful frame reception, and T_{data} is the data transmission time in the basic mode.

The worst case for SD would be when we consider high CW values, but no congestion is taking place. This is the case at $t = 150s$, when we stop all but one transmission in order to observe the remaining throughput. Fig. 1 shows that SD still behaves better than resetting the CW. After few successful transmissions, SD would reach the CW_{min} value, which CW reset scheme would have directly reached. But we notice that the CW reset scheme takes long time to increase its throughput. In fact, all traffic sources (but one) stop at $t = 150s$, but the effect is "shifted" to around $t = 168s$. This is due to the residual frames queued in the interfaces of all 49 transmitters (the interface queue length is 50 frames). After sources stop, these remaining frames will continue contending to access the channel, possibly collide and get retransmitted resulting in the following:

- Spread the sudden sources stop in time, therefore we cannot observe the real overhead of SD when traffic sources stop abruptly.
- As congestion still exists, SD still shows better performance.

Consider now the same scenario as before, but with shorter interface queue lengths (= 2), in order to remove the effect of smoothly stopping sources and observe the real overhead due to SD. Fig. 2 shows that the above queueing effects are eliminated, and the overhead due to SD can be observed at its worst (no congestion, high CW values, i.e. at $t = 150s$).

This shows that SD performs as well as CW reset scheme at low congestion, even right after high congestion. This can be considered as the response of the function to the congestion changing frequency at its maximum, i.e. when the number of contending nodes vary up and down very fast. SD performs as well at lower congestion variations, when the number of transmitting sources changes up and down more slowly.

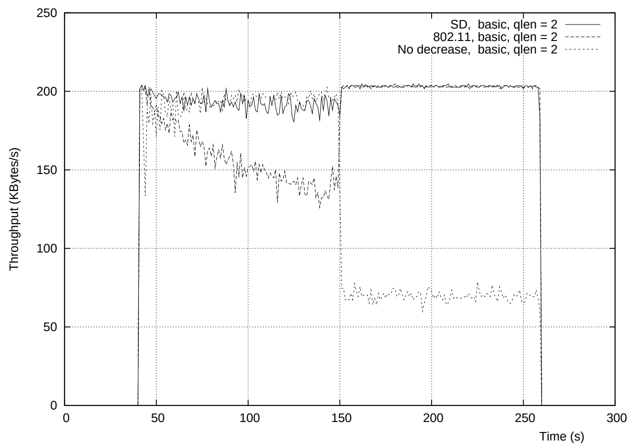

Figure 2: Total throughput comparison, without RTS/CTS, qlen=2.

For comparison convenience, we add a third curve to Fig. 2, showing the overall throughput when we do not decrease the CW at all, i.e. keeping it at its maximum reached values. This shows that the backoff time cannot be absolutely considered as negligible and must be reduced upon successful transmissions. The performance decreases considerably at low congestion and high CW values, as we can see for the remaining active flow after $t = 150s$.

Fig. 3 shows the delay observed for the same simulation scenarios. We can see how the delay increases with the number of contending nodes for both SD (solid curve) and CW reset scheme (dashed curve). SD shows lower delays and jitters. Since the CW decreases slowly, more collisions and retransmissions are avoided, which translates to lower average delays. And since the CW varies slowly, staying more adapted to the actual congestion level, the jitter is lower than the one with CW reset scheme by tens of milliseconds. The probabilities of a successful transmission change with the CW variation, therefore using sudden CW reset after each successful transmission leads to very high jitters. SD has lower jitters, showing the convenience of this approach typically at high congestion levels.

For completeness, we show in Fig. 4 the packet delays when we do not decrease the CW at all (solid curve). It has similar behavior to SD at high congestion levels. However, after the sudden congestion level drop (at $t = 150s$), this mechanism keeps high CW values leading to high delays (and low throughput, as shown in Fig. 4). These delays exist before the congestion level drop, but are in favor of collision avoid-

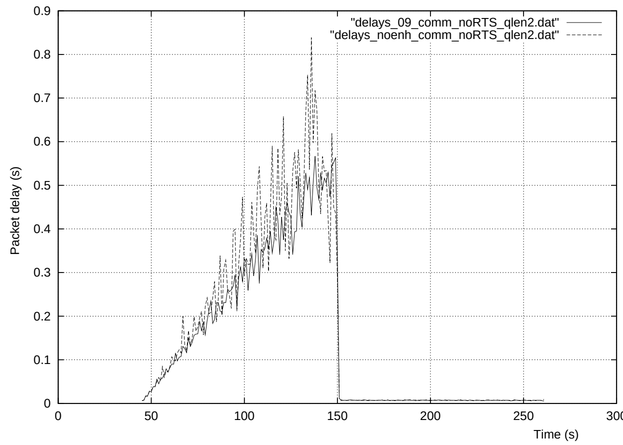

Figure 3: Packet delays comparison, without RTS/CTS, qlen = 2.

ance, increasing the throughput and lowering the overall packet delays compared to CW reset scheme. We should note that when we consider longer interface queues (e.g. 50), delays become orders of magnitude higher than the delays in Fig. 3 and 4.

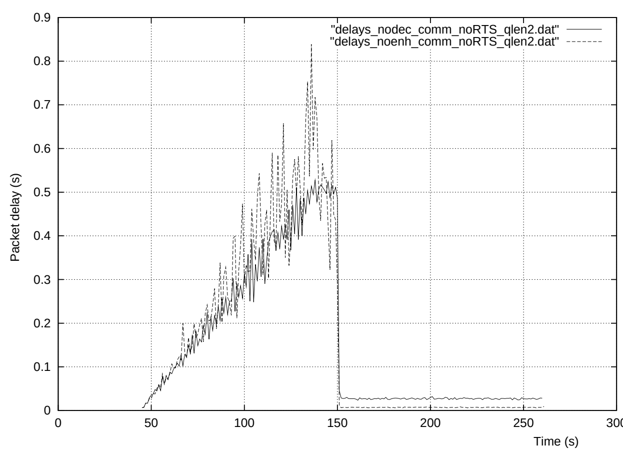

Figure 4: Packet delays comparison, without RTS/CTS, qlen = 2, the no-decrease scheme.

When we use short data frames, the relative gain decreases and SD becomes less efficient: The backoff overhead introduced by SD becomes comparable to the frame payload. To that end, consider the RTS/CTS exchange before a data frame transmission. SD avoids (short) RTS collisions which are less severe, from the data rate point of view. Therefore we observe low gain of SD over CW reset scheme.

This can be seen in Fig. 5. When congestion is low, we observe no gain, SD performs as well as CW reset scheme. At high congestion level (at $t = 150s$), we observe a better throughput enhancement. Obviously, RTS/CTS adds overhead and performs less than the basic scheme, whether using SD or CW reset scheme.

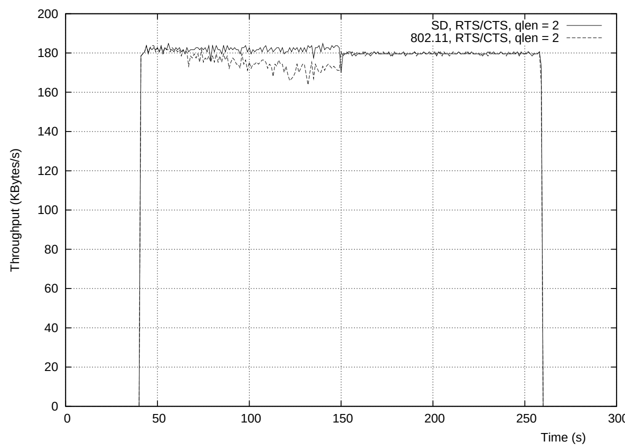

Figure 5: Total throughput comparison with RTS/CTS.

In order to evaluate the performance of the SD approach, we introduce two metrics used in feedback control theory [10]:

- *Throughput gain (G)*: This is the ratio of the throughput obtained by applying SD over the throughput obtained by applying CW reset scheme.
- *Settling time (T_l)*: After a sudden decrease of active WTs number (e.g. at $t = 150s$), T_l is the time it takes a single flow to reach its steady state throughput, with small CW values. T_l characterizes the system *response time* using CW decrease.

In the following we will use different CW decrease factors δ and different data rates λ ($\lambda =$ (source data rate)/(maximum channel capacity)) to evaluate G and

T_l. Fig. 6 shows the throughput gain G function of the CW decrease factor δ. Each point is averaged over 9 simulation runs, and the confidence interval is 95%. We can see that:

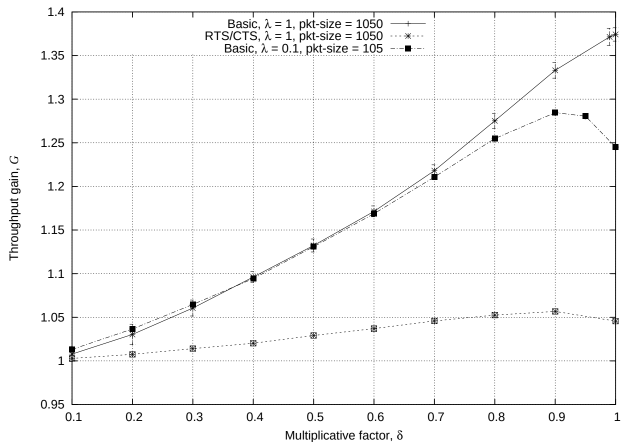

Figure 6: Throughput gain, G, vs. CW decrease factor δ.

- When δ decreases, the slow CW decrease becomes closer (resembles more) to CW reset scheme and shows no enhancement over this last, ($G \rightarrow 1$).
- However, when the multiplicative factor δ is high, CW decreases slowly upon each successful frame transmission, still avoiding future collisions and retransmissions, therefore the throughput is higher than with CW reset scheme ($G > 1$).
- When using small frame sizes, gain decreases since collisions are less severe, and the maximum gain G_{max} is around $\delta_{max} = 0.9$. Beyond this value the backoff payload becomes considerable relative to the frame size and G decreases.
- As λ decreases, the throughput gain G decreases for all values of δ. In fact, when data rates decrease, we observe fewer collisions leading to fewer CW increase and CW decrease. Therefore the gain of SD over CW reset scheme gets lower and converges to unity.
- When $\delta = 1$, we observe a non-negligible gain $G > 1$ when the channel is highly congested (as seen in Fig. 2). However, when the channel suddenly becomes less congested, the CW value keeps constantly high, increasing overhead,

and decreasing throughput efficiency. For low data rates, this overhead (when $\delta = 1$) is negligible relative to the idle channel periods between consecutive packets. Therefore the gain $G = 1$. However, when $\lambda = 1$, this overhead becomes considerable leaving large idle gaps between packets, reducing efficiency, therefore the gain drops to $G = 0.48$.

- When using $\delta < 1$, the CW size (and overhead) will progressively decrease upon each successful transmission. Therefore the overhead cited above (with $\delta = 1$) still exists but for a transient period only, the duration of which is function of δ, the frame data rate λ and the corresponding successful transmissions. This transient period is characterized by T_l, the settling time we defined above, and will be analyzed in section 4.

3.1 Mathematical model

Our analysis is divided into two parts. First we study the behavior of a single mobile station with a SD Markov model, and we compute the stationary probability τ that the station transmits a packet in a randomly chosen time slot. This probability does not depend on the access mechanisms (with or without RTS/CTS scheme). Second, by studying the events occurring within a time slot, we express the channel throughput as a function of τ with and without RTS/CTS schemes. We get then a system of two equations that we solve for the channel throughput by getting rid of τ.

3.1.1 Analysis of packet transmission probability

We make the same assumptions as in [11]. A fixed number n of contending stations is considered and the transmission queue of each station is always nonempty. Each packet has to wait for a random backoff time decrement to zero before transmitting. The time slot duration is defined as σ, and p denotes the probability that a packet collides. A time slot is equal to 802.11 time slot σ_0 if no packets are transmitted. If a packet is being transmitted, σ is equal to the busy period until the channel is idle again for a time period equal to DIFS. We define two stochastic processes to model the protocol behavior, see Fig. 7. First, $b(t)$ represents the backoff counter of the time a station has to wait before it can transmit. This process has the range from 0 to the current CW size. Another stochastic process $s(t)$ is defined as the backoff stage at different CW level. $s(t)$ scales from 0 to m, with m being the maximum CW stage.

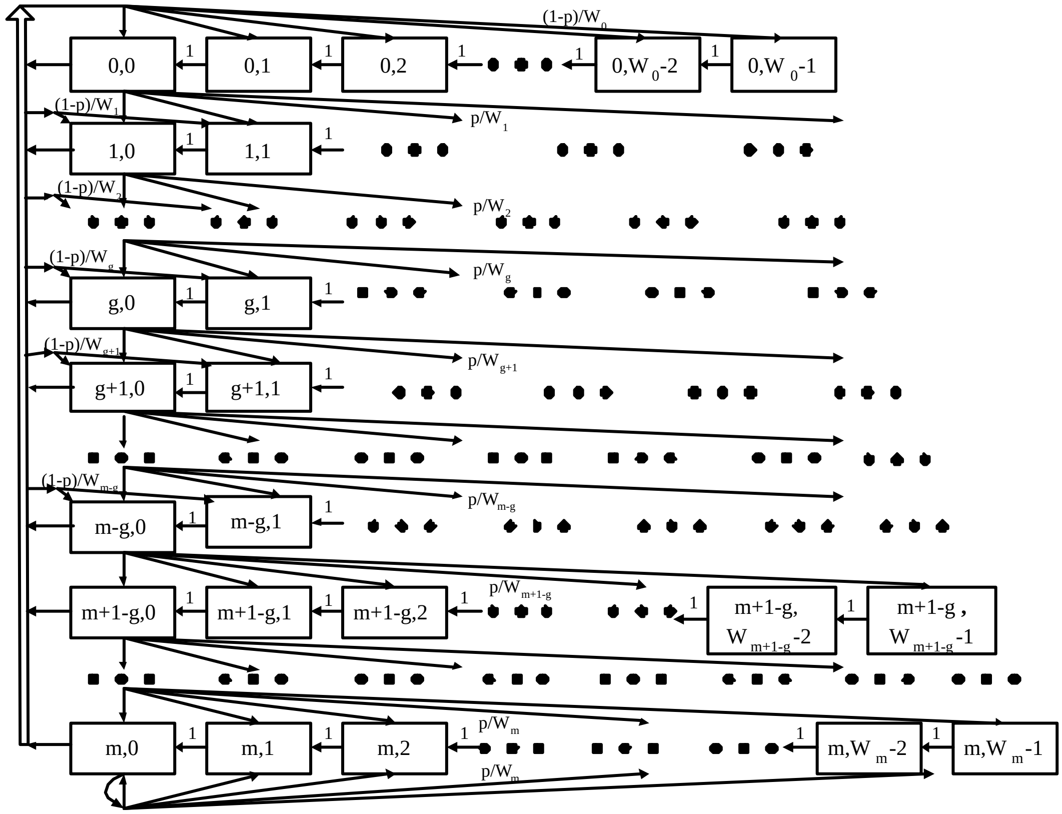

Figure 7: Markov chain model for SD scheme.

With these assumptions, the bi-dimensional stochastic process $\{s(t), b(t)\}$ fulfills the properties of an homogeneous discrete Markov chain. The Markovian property does not hold for the process $b(t)$ alone, which is dependent on the backoff stage history. For simplicity, we write W_i instead of CW_i and W_0 instead of CW_{min}. Since the contention window doubles after each collision, we can write $W_i = 2^i \times W_0$, where $0 \leq i \leq m$. The maximum backoff stage m is the value such that $CW_{max} = 2^m \times W_0$. We suppose that the constant decrease factor δ has a power of two form $\delta = 1/(2^g)$, where the constant factor g is an integer with $g > 0$. This choice of δ limits the number of states of the Markov chain and simplifies the analysis, without impacting the results. Thus, the new CW value when a packet is correctly transmitted will be:

$$CW_{new} = \max(W_0, \delta \times W_i) = \max(W_0, 2^{i-g} \times W_0)$$
$$= \max(W_0, W_{i-g}).$$

Consider the transitions of the SD scheme between time slots. For instance, we ignore time slots where the station is transmitting. Fig. 7 explains the behavior of the Markov chain. The only non-null one-step transition probabilities are:

$$\begin{array}{l} P\{i,k|i,k+1\} = 1, \text{for } k \in [0, W_i - 2], i \in [0, m]. \\ P\{0,k|i,0\} = (1-p)/W_0, \\ \qquad\qquad\qquad \text{for } k \in [0, W_0 - 1], i \in [0, g-1]. \\ P\{i-g,k|i,0\} = (1-p)/W_{i-g}, \\ \qquad\qquad\qquad \text{for } k \in [0, W_{i-g} - 1], i \in [g, m]. \\ P\{i,k|i-1,0\} = p/W_i, \\ \qquad\qquad\qquad \text{for } k \in [0, W_i - 1], i \in [1, m]. \\ P\{m,k|m,0\} = p/W_m, \qquad \text{for } k \in [0, W_m - 1]. \end{array} \tag{1}$$

The first equation in (1) accounts for the fact that the backoff timer has not reached 0 and that it is decremented by 1 at the beginning of each time slot. The second and third equations are specific to the SD scheme. The second equation accounts for the fact that when $\delta \times W_i$ is smaller than W_0, we reset W_i to W_0, and a new backoff is uniformly chosen in the range $(0, W_0 - 1)$. The third equation accounts for the fact that when $\delta \times W_i$ is larger than W_0, we decrease W_i slowly to the new value W_{i-g} and we choose the new backoff counter randomly in the range $(0, W_{i-g})$. The fourth and the fifth equations correspond to the cases where a collision occurs.

Let $\pi_{i,k} = \lim_{t\to\infty} P\{s(t) = i, b(t) = k\}$, $i \in [0, m], k \in [0, W_i - 1]$, be the stationary distribution of the Markov chain. As the Markov chain is ergodic, this distribution exists and is unique. First, we will express all $\pi_{i,k}$ as function of $\pi_{0,0}$, then we will use the normalization equation to solve for $\pi_{0,0}$, and hence for all $\pi_{i,k}$.

From the Markov chain above, we can see that the incoming traffic to stage i from either $(i+g, 0)$ after a successful transmission, or from $(i-1, 0)$ after a collision, is uniformly distributed over all possible backoff values at this stage. Afterwards, the counter is decremented by one and finally reaches $(i, 0)$. So, the stationary probability $\pi_{i,0}$ is given by:

$$\begin{array}{lll} \pi_{0,0} & = (1-p)\sum_{j=0}^{g} \pi_{j,0}. & \\ \pi_{i,0} & = p\,\pi_{i-1,0} + (1-p)\pi_{i+g,0}, & 0 < i \leq m-g. \\ \pi_{i,0} & = p\,\pi_{i-1,0}, & m-g < i < m. \\ p\,\pi_{m-1,0} & = (1-p)\pi_{m,0}. & \\ \Rightarrow \pi_{m,0} & = \frac{p}{1-p}\,\pi_{m-1,0}, & i = m. \end{array} \tag{2}$$

The first equation in (2) accounts for the fact that stage 0 can only be reached from stages $j \leq g$ in the SD scheme, the stages $j > g$ can not directly decrease to stage 0. The second equation in (2) says that for stages $0 < i \leq m-g$, there are two different inputs: From the previous stage with collision probability p and from the stage $i+g$ after a successful transmission with a probability $1-p$. For stages $i > m-g$, there will be no input from stages $i+g$, because $i+g$ is bigger than the maximum stage number m. For stage m, we fall into a special case, since after a collision the contention window remains at this stage.

Now, according to the Markov chain regularities, for each $k \in [1, W_i - 1]$, $\pi_{i,k}$ can be written as:

$$\pi_{i,k} = \frac{W_i - k}{W_i} \begin{cases} (1-p)\sum_{j=0}^{g} \pi_{j,0}, & \text{for } i = 0. \\ p\,\pi_{i-1,0} + (1-p)\,\pi_{i+g,0}, & \\ \qquad \text{for } 0 < i \leq m-g. & \\ p\,\pi_{i-1,0}, \text{ for } m-g < i \leq m. & \\ p\,(\pi_{m-1,0} + \pi_{m,0}), \text{for } i = m. & \end{cases} \tag{3}$$

The ratio before the parentheses accounts for the distribution of probabilities for each state in a stage. When we move in a stage to the right, the probability decreases by $1/W_i$, since we do not get the input of the previous state in the same stage. Thus, we can obtain the relationship between $\pi_{i,k}$ and $\pi_{i,0}$: $\pi_{i,k} = (W_i - k)/W_i \times \pi_{i,0}$. By using (2), we get the term on the right-hand side of the parentheses in (3). Equation (3) allows then to compute all stationary probabilities as a function of $\pi_{0,0}$ and p. Obtaining closed-form expressions does not seem possible, so we proceed by solving the system numerically with Matlab: First we solve formulas in (2) to obtain $\pi_{i,0}$ that are only dependent on $\pi_{0,0}$ and p. Then we plug them into (3) to obtain $\pi_{i,k}$ that are only dependent on $\pi_{0,0}$ and p. $\pi_{0,0}$ is finally computed by using the normalization condition:

$$1 = \sum_{i=0}^{m} \sum_{k=0}^{W_i - 1} \pi_{i,k}. \tag{4}$$

We compute now τ, the probability that a station transmits in a time slot. This probability is simply the sum of probabilities of all $(i, 0)$ states,

$$\tau = \sum_{i=0}^{m} \pi_{i,0} = f(p, W_0, g, m). \tag{5}$$

This expression of τ is a function of p, which is unknown. The other three variables (W_0, g, m) have known values. Let us assume independence of all stations sharing the medium, i.e. the probability that a station encounters a contention is independent of the status of the other stations. The n stations are identical so they all transmit packets in a slot time with the same probability τ. Consider that a station transmits a packet in a time slot. p is then the probability that at least one other station transmits a packet in the same slot:

$$p = 1 - (1 - \tau)^{(n-1)}. \tag{6}$$

We obtain a non-linear system of two equations (5) and (6), that can be solved for p and τ. This system certainly has a solution, since (i) the expression of p in (6) is continuously increasing with τ, with $p = 0$ for $\tau = 0$ and $p = 1$ for $\tau = 1$, and (ii) the expression of τ in (5) is continuous with p. A sufficient condition for this solution to be unique is that the expression of τ in (5) is continuously decreasing with p, i.e. more contention leads to less transmissions. Our numerical results show that this is always the case and hence a unique solution for our model always exists.

3.1.2 Throughput

Denote by S the per station throughput, which is by definition the average volume of data correctly transmitted by a station in a slot time divided by the average slot time duration. Consider a random time slot, let P_{tr} be the probability that there is at least one transmission in this time slot, and let P_s be the probability of one successful transmission given that there is at least one transmission. Note that

$$P_{tr} = 1 - (1 - \tau)^n, \text{ and } P_s = \frac{n\tau(1-\tau)^{n-1}}{1 - (1-\tau)^n}.$$

$$\text{Hence, } S = \frac{P_{tr} P_s E[P]}{(1 - P_{tr})\sigma + P_{tr} P_s T_s + P_{tr}(1 - P_s) T_c} \tag{7}$$

where T_s is the average time the channel is sensed busy because of a successful transmission, and T_c is the average time the channel sensed busy because of a collision. We use in our analysis the values of T_s and T_c computed in [11]. Note that the throughput expression (7) does not specify the access mechanism employed. To account for whether RTS/CTS is used or not, we only need to specify the corresponding values T_s and T_c [11].

Fig. 8 shows the throughput model and simulation results for various decrease factors (δ) and for legacy 802.11, when we increase the number of contending nodes (basic scheme, 1050-bytes packets, 1 Mbps channel). The model results are quite similar to simulation results. We can see the considerable throughput enhancement we get with high values of δ and high number of contending nodes. SD throughput gain decreases when the number of contending nodes decreases and when δ decreases, but it keeps outperforming the legacy 802.11.

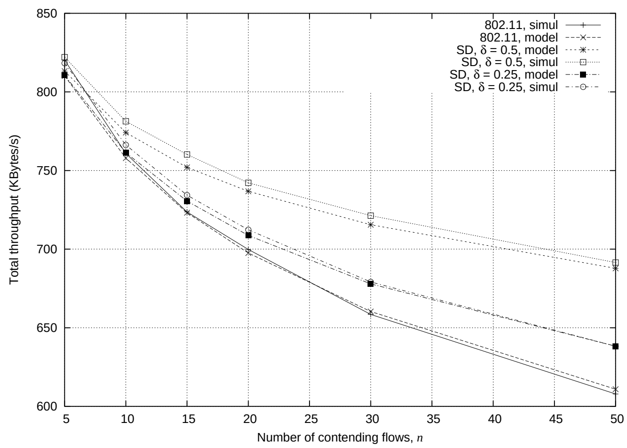

Figure 8: Throughput of SD and 802.11 vs. the number of contending flows, model and simulation results.

Fig. 9 shows the throughput gain of SD over 802.11 when varying the CW_{min} values and the number of contending nodes. Simulation and model show close results: The gain decreases when CW_{min} increases, since increasing CW_{min} contributes to collision avoidance, hence the effect of SD decreases. Furthermore, as cited before, this gain increases with the number of contending flows.

4 Settling time T_l

To measure T_l with acceptable precision, we cannot proceed as in Fig. 1 right after all traffic flows (except one) stop and simply measure the time the remaining flow

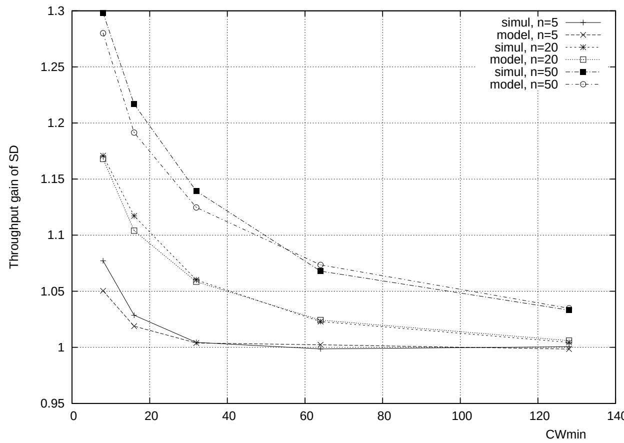

Figure 9: Throughput gain of SD over 802.11, vs. CW_{min}, varying the number of contending flows.

takes to recover a stable state. Since this last is contending to access the channel with other flows, it has a non-zero probability to access the channel and reset its CW right before $t = 150s$ in Fig. 2, and therefore start its transient period with a short CW, unsuitable to measure T_l.

We proceed using a different simple scenario[1]: A single flow is considered. We force the CW to its maximum, 1023, as it would be in highly congested environments. This reduces its throughput considerably. Then we let the CW use SD and CW reset scheme respectively, and measure the settling times T_l.

Fig. 10 shows that, as expected, when δ increases, we need more successful transmissions before throughput reaches its steady state, that is T_l increases. This increase is much higher than linear, especially for high δ values. The reader should distinguish the settling time T_l, which concerns throughput stability, from frame transmission delays. In the previous examples, a T_l of $100ms$ simply means that 40 consecutive frames should be sent successfully before the throughput reaches its high steady state. However, evaluating the user perception of T_l is out of scope of this work.

Choosing the right multiplicative decrease factor δ is a compromise between having a high throughput gain G and a short settling time T_l, for the case of sudden

[1]This scenario corresponds to the system response to an impulse input, from the feedback control point of view.

congestion decrease. Intermediate δ values like 0.6-0.9 would satisfy such a tradeoff. For smoother congestion decrease[2], one would choose higher δ values to get higher throughput gains, without much caring about T_l.

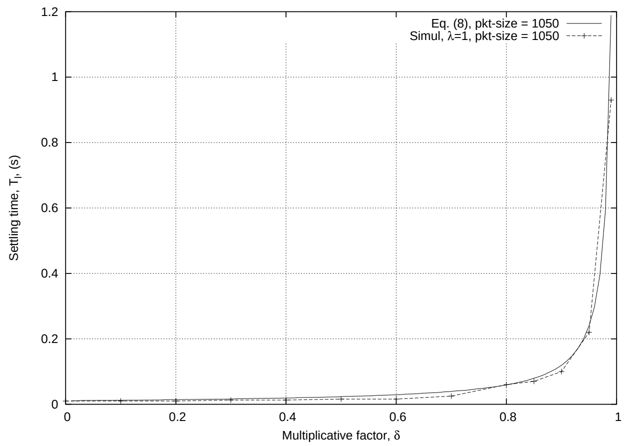

Figure 10: Settling time T_l vs. δ.

We also investigated linear SD which showed it can reach the same gain values as multiplicative CW decrease. However, the settling time T_l is higher than with multiplicative CW decrease, especially for small linear decrease constants (α) which would result in good throughput enhancement.

Finally we should note that in [5], the authors use linear SD with $\alpha = 1$. This surely enhances throughput, as very high δ values do with multiplicative SD. However, very high δ values and very low α values would lead to unacceptable settling times T_l, if one considers sudden congestion level drop. From the user point of view, high settling time values (T_l) mean longer delays before the user gets the maximum throughput after moving from a highly congested area to a low congested one, or when all of his neighbors suddenly stop their transmissions.

It is easy to obtain a closed-form expression for the maximum settling time T_l. We need to send l consecutive frames successfully, to reach the "optimal" throughput (with CW_{min}), i.e.

$$CW_{max} \times \delta^l = CW_{min},$$

[2] Practically, this is hard to predict.

therefore $l = \lfloor \frac{\ln(CW_{min}/CW_{max})}{\ln(\delta)} \rfloor$, those l frames take

$$T_l = \sum_{i=0}^{l}(T_s + O_{bkof}(i)) = (l+1)T_s + \frac{CW_{max}}{2}\sigma\frac{1-\delta^{l+1}}{1-\delta}, \tag{8}$$

where i is the transmission attempt number, T_s is the frame transmission time with its corresponding $DIFS$, $SIFS$ and ACK transmission time, and σ is the time slot duration.

5 Fairness analysis

This section is divided into two parts. The first one analyzes short-term and long-term fairness of SD. The second one analyzes long-term fairness between 802.11 nodes and SD nodes operating together.

5.1 Fairness amongst similar nodes

Before discussing the fairness of the SD scheme, let us check some issues related to fairness in legacy 802.11. To measure fairness, we use Jain's index of fairness (F_J)[12]. We consider a given number of accesses (a window) to the channel and compute F_J as:

$$F_J = \frac{(\Sigma_{i=1}^{n}\gamma_i)^2}{n\Sigma_{i=1}^{n}\gamma_i^2} \tag{9}$$

where n is the number of nodes and γ_i is the proportion of successful accesses of node i during the considered window. F_J is equal to unity when all nodes equally share the medium, and is equal to $1/n$ when a single node monopolizes the channel (in which case $F_J \to 0$ when $n \to \infty$). We compute the average F_J by sliding the window through all the simulation time. Fig. 11 shows, as in [13], the weak fairness of 802.11 on the short-term scale. This fairness obviously improves when the window size used for measurement gets bigger.

When we increase the CW_{min} value, we see that fairness also improves (Fig. 11): After a successful transmission, a node (with a high CW_{min}) has a lower probability to access the channel right afterwards, which gives other contending nodes higher probabilities to access the channel, and hence improves the fairness. However, this is not the case when we increase the number of contending nodes (Fig. 12):

Indeed, when we increase the number of contending nodes, we increase the collision rate. This increases the risks of having nodes with high CWs (after collisions),

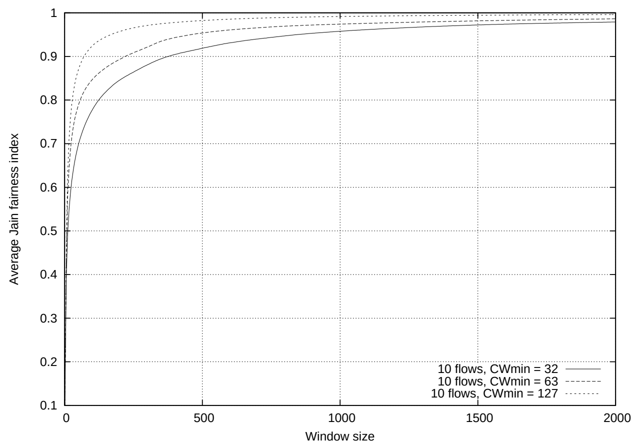

Figure 11: Fairness of 802.11 when varying CW_{min}

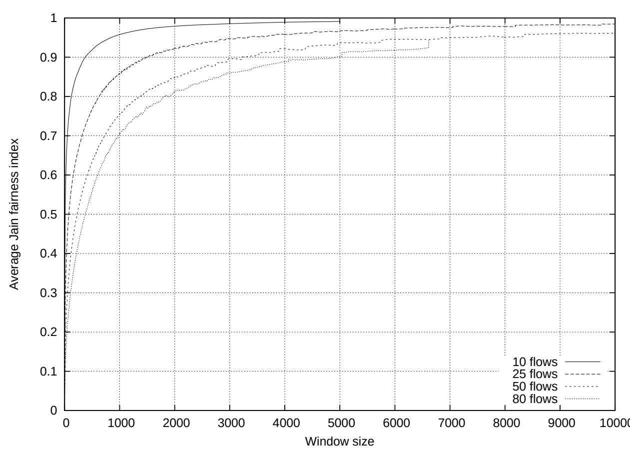

Figure 12: Fairness of 802.11 when varying the number of contending nodes

while others get the chance to transmit several frames more frequently, therefore degrading fairness.

The above two aspects, of improving fairness with CW_{min} and degrading it with the number of contending nodes, are combined when we increase the number of nodes with SD (Fig. 13).

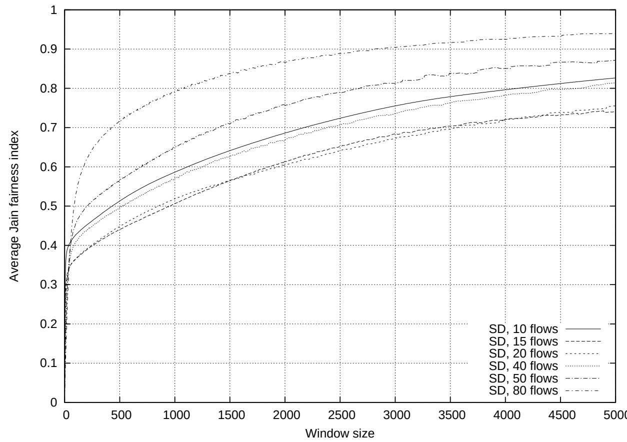

Figure 13: Fairness of SD when varying the number of contending nodes

Using SD increases the average CW sizes, which is supposed to improve fairness. However the increasing number of nodes tends to degrade fairness. Therefore, with a fixed multiplicative decrease factor (0.9), we notice that when we increase the number of flows, fairness decreases down to a given level, then starts increasing. That is the point where large CW sizes compensate the unfairness of the high number of contending flows.

Fig. 14 compares the fairness of 802.11 and SD: For a small number of contending flows, 802.11 is fairer than SD. When we increase the number of flows, the fairness curves of the two schemes get close. For high number of contending flows, SD shows more fairness than 802.11.

5.2 Fairness with legacy 802.11 nodes

The main drawback of using SD is the unequal share of data rate it gets when it coexists with 802.11. Consider the scenario where part of the competing nodes use 802.11 and the other part uses SD.

Let n_1 be the number of 802.11 stations, τ_1 be the probability that a 802.11 station transmit in a time slot, p_1 be the collision probability seen by an 802.11 station, P_{tr1} be the transmission probability that one 802.11 transmission in the considered time slot, it is the same as τ_1. P_{s1} be the probability that one 802.11 station transmission occurring on the channel is successful. Let $n_2, \tau_2, p_2, P_{tr2}$ and

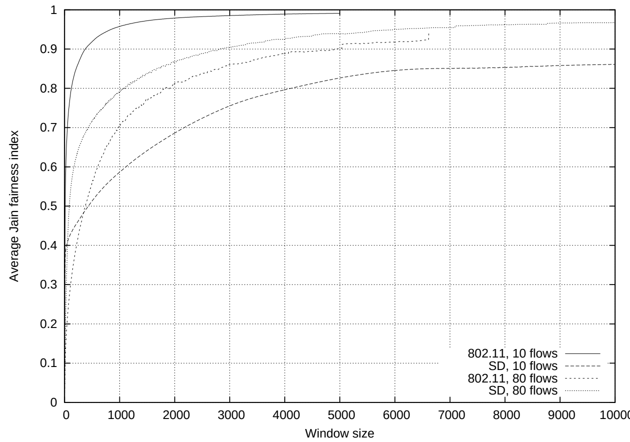

Figure 14: Comparing fairness of 802.11 and SD

P_{s2} be the corresponding values for SD stations. We keep the same meanings and notations of T_s, T_c, P_{tr}, and P_s as in 3.1.2, the throughput of one 802.11 station will be:

$$S_1 = \frac{P_{tr1} P_{s1} E[P]}{(1 - P_{tr})\sigma + P_{tr} P_s T_s + P_{tr}(1 - P_s) T_c}.$$

The throughput of a SD station will be:

$$S_2 = \frac{P_{tr2} P_{s2} E[P]}{(1 - P_{tr})\sigma + P_{tr} P_s T_s + P_{tr}(1 - P_s) T_c},$$

where:

$$\begin{aligned} P_{tr1} &= \tau_1 \\ P_{s1} &= (1 - \tau_1)^{(n_1 - 1)} \times (1 - \tau_2)^{n_2} \\ P_{tr2} &= \tau_2 \\ P_{s2} &= (1 - \tau_1)^{n_1} \times (1 - \tau_2)^{n_2} \\ P_s &= \frac{n_1 \tau_1 P_{s1} + n_2 \tau_2 P_{s2}}{P_{tr}} \end{aligned}$$

In the following we keep the total number of 802.11 and SD stations to a fixed value of 10 or 20. Fig. 15 shows the throughput of an 802.11 node and an SD node when the proportion of 802.11 nodes varies, based on the above model and the simulation results.

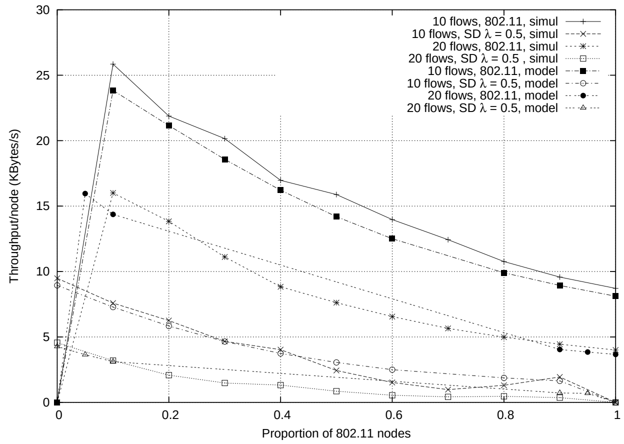

Figure 15: 802.11 and SD nodes working together

Nodes using SD have high CW values, trying to avoid collisions at high congestion levels. This is not the case of 802.11 which have relatively smaller CWs and keep severe contention and less collision avoidance. Obviously, this results in unequal share of the available data rate, to the advantage of 802.11 nodes, over all congestion levels (i.e. number of flows). The throughput gain of an 802.11 node over an SD one decreases when the total number of nodes increases: High CW values SD nodes have become more useful to avoid collisions and increase the obtained data rate, but 802.11 nodes still have the biggest share of the data rate (gain > 1).

The previous results show that it is better not to use SD as soon as one regular 802.11 station is present. The following mechanism can be used to decide to use SD or not. In case of infrastructure mode, each SD station informs the AP that it is SD-compliant (using an extended *Probe Response Frame*), then the AP can decide based on current information received if SD mode can be used or not in the next beacon interval. In other words, stations use SD only if explicitly mentioned by the AP in the beacon (denoted by extended beacon format). In case of ad-hoc mode, beacon generation is distributed between each station. If one station does not send an extended beacon format, then all the SD stations will decide to switch back to 802.11 operation mode but they will continue to report their SD compliance in their beacon. If no more regular beacon is received after some timeout, SD stations can decide to use SD again. Actually the use of the beacon to send a specific option has already been proposed in the standard to force IEEE 802.11g back to 802.11b in a mixed 802.11b/g environment.

6 Energy saving

When the congestion level is high, frames are most likely to collide and be retransmitted before reaching their destinations successfully. The energy consumption at the sender, as well as at the receiver, is therefore proportional to the number of retransmissions. Slowly decreasing CWs, as in our scheme, reduce the risks of collisions and the corresponding retransmissions, for the same number of successful receptions, saving considerable energy. We simulate a scenario where we have n flows each with 1 MBytes of data to transfer (using FTP/TCP), without RTS/CTS, considering that transmission power is 600mW and receiving power is 300mW. The average energy per successfully received bit is shown in Fig. 16.

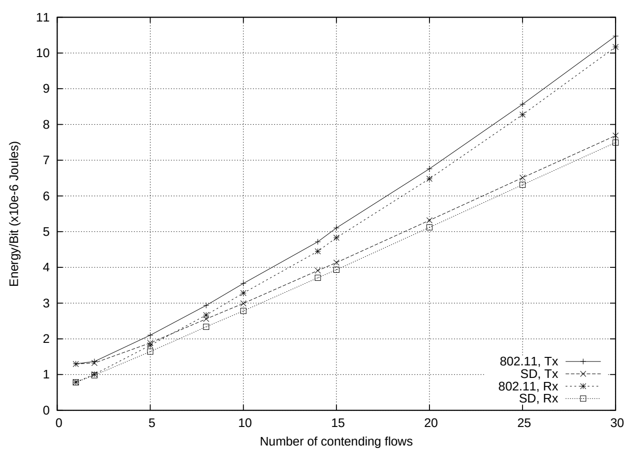

Figure 16: Throughput energy consumption

The energy curves shown here are an image of the number of retransmissions for each successfully received frame. When the number of contending flows increases, collision rate increases, spending more energy to deliver a good frame. However this is considerably lower with SD than with 802.11, due to the better adapted CW values. At low number of contending flows, no considerable collisions occur, therefore the energy consumption per successfully received packet is the same for 802.11 and for SD.

Reducing the number of retransmissions, by avoiding collisions, not only reduces the energy consumption but the total data transfer duration too. The overhead introduced by SD becomes negligible when collisions and retransmissions occur. Fig. 17 shows that, when 30 flows contend to transmit 1 MBytes each, it takes 270 seconds for 802.11 to achieve the transfer. It takes considerably less (200 seconds) for SD to do the same job.

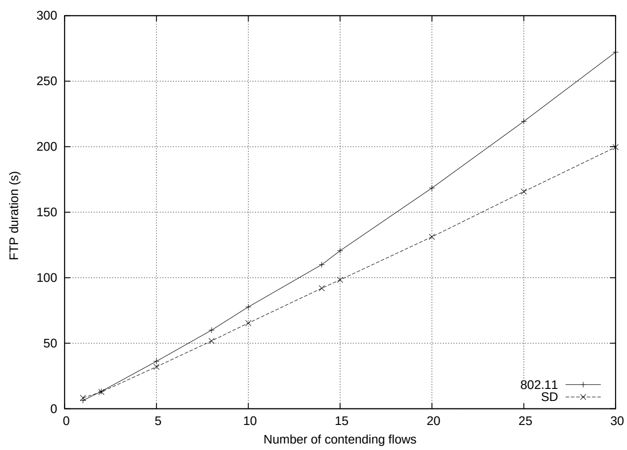

Figure 17: FTP duration comparison

This duration difference between SD and 802.11 does decrease with the number of contending flows. We should note that at higher number of contending flows, we start to observe long TCP timeouts for some flows, causing disconnections, and reconnections at later times, whether for SD or 802.11. This makes the FTP duration measurement considered here inappropriate for very high number of flows.

7 Noisy channels

802.11 and SD both suffer from the same problem in noisy channels: They cannot distinguish noise lost frame from collision lost frames. In both situations a node does not receive its frame ACK and doubles its CW to avoid further collisions, which is not needed if the frame was noise dropped. This adds an overhead which, in addition to the noise dropped frames, reduces the throughput considerably. This can be seen in Fig. 18.

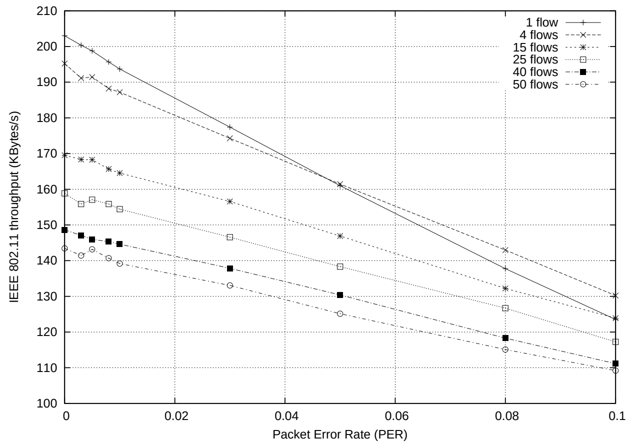

Figure 18: Throughput of 802.11 in noisy channel

For all values of packet error rate (PER) we can see that the throughput drop is much larger than the corresponding PER, because of the useless CW increase caused by noise. For instance at $PER = 0.1$, for a single flow, the throughput drop is (203007 to 123522) 39% while only 10% of the packets are corrupted[3].

The effect of noise lost frames is even more harmful to SD since it causes CWs to get high, without necessarily avoiding any collisions. More precisely, both 802.11 and SD would increase the same way, but SD CWs decrease slower, adding more overhead, therefore wasting bandwidth, unless collision also exists.

Fig. 19 shows the throughput gain of SD over 802.11 when varying the PER and the number of contending flows, without RTS/CTS.

For a single flow accessing the channel, the throughput of SD stays close to that of 802.11 as long as the PER is < 0.01. Beyond this point (very severe channel conditions) the frequently corrupted frames keep the CWs relatively high with SD, and the gain decreases. The gain increases with the number of competing flows. At high PER, frequently corrupted frames still cause CWs to stay high, but to the advantage of avoiding collisions in this case.

[3]For the case of four flows in Fig. 18, at $PER > 0.05$, we can see that the throughput is higher than that of a single flow. This is due to the fact that the aggregated throughput of the flows, reduced by 10%, observe no considerable collisions that may reduce the throughput.

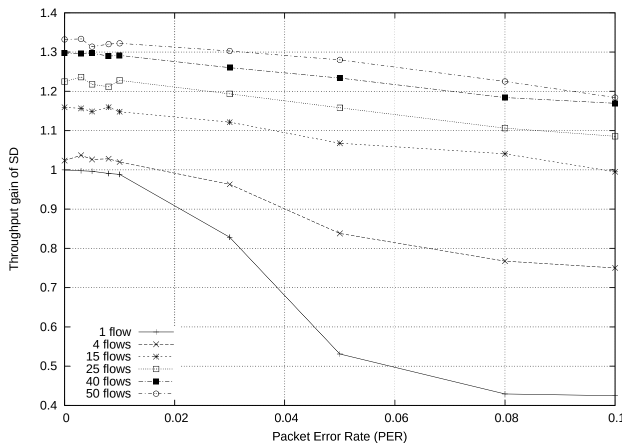

Figure 19: Throughput gain of SD over 802.11 in noisy channel

8 Conclusion

In this paper we investigated slow CW decrease (SD) instead of CW reset after each successful frame transmission. This would avoid future collisions, considering that congestion level is unlikely to drop suddenly. It also reduces the number of frame retransmissions (which would also reduce congestion on the channel), increasing the throughput considerably, decreasing delays and jitters. It performs as well as IEEE 802.11 in non-congested environments, and shows considerable gain over the latter in congested ones. The throughput gain is function of frame lengths and data rates. We showed, using simulations and mathematical modeling, the considerable gain when using large data frames (37%), and extended the analysis for the worst gain values, that is for short data frames, e.g. when using RTS/CTS. Multiplicative CW decrease functions showed high throughput gains, with relatively low settling times after sudden congestion level drops. Fairness and coexistence between SD and 802.11 are also explored, showing the weak points and their solutions as well. Last, the effect of channel noise on SD and its considerable power saving are analyzed. Future work includes adaptive CW decrease algorithms, in which decrease parameters change with the congestion load level in order to further enhance the fairness of SD.

References

[1] Brian P. Crow, Indra Widjaja, Jeong Geun Kim, and Prescott T. Sakai, "IEEE 802.11 wireless local area networks," *IEEE Communication magazine*, September 1997.

[2] Bob O'Hara and Al Petrick, *IEEE 802.11 handbook. A designer's companion.*, IEEE Press, 1999.

[3] LAN/MAN Standards Committee, *ANSI/IEEE Std 802.11:Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Computer Society, 1999.

[4] L. Kleinrock and F. A. Tobagi, "Packet Switching in Radio Channels: Part 1: CSMA Modes and Their Throughput Delay Characteristics.," *IEEE Trans. on Comm. COM-23*, 1975.

[5] Vaduvur Bharghavan, Alan Demers, Scott Shenker, and Lixia Zhang, "MACAW: A media acess protocol for wireless LANs.," in *Proceedings of ACM Sigcomm*, 1994.

[6] Thyagarajan Nandagopal, Tae-Eun Kim, Xia Gao, and Vaduvur Bharghavan, "Achieving MAC layer fairness in wireless packet networks.," in *Proceedings of Mobicom*, 2000.

[7] Timucin Ozugur, Mahmoud Naghshineh, Kermani Parviz, Michael Olsen, Babak Rezvani, and John Copeland, "Balanced media acess methods for wireless networks.," in *Proceedings of Mobicom*, 1998.

[8] Mo Jeonghoon, J. La Richard, Anantharam Venkat, and C. Walrand Jean, "Analysis and comparison of TCP Reno and Vegas.," in *Proceedings of Infocom*, 1999.

[9] D. Chiu and R. Jain, "Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks.," in *Computer Networks and ISDN Systems 17*, 1989.

[10] Charles L. Phillips and Royce D. Harbor, *Feedback control systems*, Prentice-Hall, 1988.

[11] Giuseppe Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE journal on selected areas in communications*, vol. 18, 2000.

[12] Rajendra K. Jain, Dah-Ming W. Chiu, and William R. Hawe, "A quantitative measre of fairness and discrimination for resource allocation in shared computer system," Tech. Rep., Digital Equipment Corporation, 1984.

[13] Can Emre Koksal, Hisham Kassab, and Hari. Balakrishnan, "An analysis of short-term fairness in wireless media access protocols," in *Proceedings of ACM Sigmetrics*, 2000.

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399